A Sequential Linear Programming Algorithm for Economic Optimization of Hybrid Renewable Energy Systems

M. Vaccari[a], G.M. Mancuso[a], J. Riccardi[b], M. Cantù[c], G. Pannocchia[a,*]

[a] *University of Pisa, Department of Civil and Industrial Engineering, Largo L. Lazzarino 2, 56126 Pisa, Italy*
[b] *Enel Green Power, Innovation and Sustainability, Via Andrea Pisano 120, 56120 Pisa, Italy*
[c] *Enel Engineering and Research, Via Andrea Pisano 120, 56120 Pisa, Italy*

Abstract

Combining renewable energy sources, as photovoltaic arrays (PV), wind turbine (WT), biomass fuel generators (BM), with back-up units to form a Hybrid Renewable Energy System (HRES) can provide a more economic and reliable energy supply architecture compared to the separate usage of such units. In this work an optimization tool for a general HRES is developed: it generates an operating plan over a specified time horizon of the setpoints of each device to meet all electrical and thermal load requirements with possibly minimum operating costs. A large number of devices, such as conventional and renewable source generators, mandatory and deferrable/adjustable electrical loads, batteries, combined heat and power configurations are modeled with high fidelity. The optimization tool is based on a Sequential Linear Programming (SLP) algorithm, equipped with trust region, which is able to efficiently solve a general nonlinear program. A case study of a real HRES in Tuscany is presented to test the major functionalities of the developed optimization tool.

Keywords: Energy systems, numerical optimization algorithms, Sequential Linear Programming, Hybrid Renewable Energy Systems (HRES)

1. Introduction

Nowadays, a large portion of the energy requirements all around the world is still supplied from conventional energy sources like coal, natural gas, crude oil, etc. On the other hand, the gradual scarcity of conventional energy resources, fuel price fluctuations and harmful emissions have made power generation by conventional methods only, unsustainable and non-viable on the long term. A possible solution can be found in the use of renewable energy sources (i.e., solar, hydroelectric, biomass, wind, ocean and geothermal). Each one has its own special advantages that make it uniquely suited to certain applications. The major drawback of the mentioned energy options is their unpredictable nature and dependence on weather and climatic conditions. This problem can be overcome by integrating renewable and traditional resources in a suitable hybrid architecture. Hybrid Renewable Energy Systems (HRES) are composed of one renewable and one conventional energy source or more than one renewable with or without conventional energy sources, which operate in stand alone or grid connected mode [1]. These HRES comprise a number of *devices* which may generate, absorb or store electricity and/or heat. Despite cases where the energy exchange is not possible, e.g. island operations or remote regions, the HRES is generally assumed bidirectionally interlaced with the electrical grid. In this way any electrical power generation excess/lack can be sold to/bought from the grid. On the other hand, any heat requirement, generally transported by either hot or cold media (usually water streams), has to be fulfilled in the exact amount within the HRES.

The main goal of this work is to build an optimization system that, given an HRES with all devices sized, optimizes their setpoints in order to minimize the overall operational cost over a specified time horizon. This horizon lasts usually 24 hours, but it can be longer or shorter as desired. The system is designed to meet four of the possible tariff regimes actually in force in Italy, but its structure is sufficiently general to be adapted to other energy price policies.

This paper is organized as follows. A literature review on HRES generalities and optimization methods is presented in Section 2. The HRES modeling and how its operational cost is calculated are presented in Section 3. The optimization problem is then formulated and all constraints are explained in Section 4. Based on this problem, the developed optimization algorithm is presented in Section 5. The algorithm is then tested over a real case study of an HRES located in Tuscany. Results and discussions are reported in Section 6. Finally, Section 7 summarizes the main achievements of this work.

2. Background

2.1. HRES generalities

An important feature of HRES is to combine two or more renewable power generation technologies to make best use of their operating characteristics. In this way efficiencies higher

*Corresponding author. Email: gabriele.pannocchia@unipi.it

than those obtained from a single energy source can be obtained. HRES can address limitations in terms of fuel flexibility, efficiency, reliability, emissions and economics [2]. As mentioned, an HRES can be configured either in stand-alone or in grid-parallel application modes. Selection of the application mode depends on several factors such as grid availability, cost of grid supplied electricity, and meteorological conditions in the application site.

- "On grid": there is only one link with the grid per each HRES denominated "Point of Distribution": it allows a bi-directional power flow. This is mostly used in urban sites as well as for large wind and solar farms.
- "Stand-alone": conceptually it can be obtained by a grid-parallel system, simply switching off the connection with electrical grid. Of course the starting grid-parallel system has to be equipped with back-up units and fuel generator. Stand-alone HRES are considered as one of the most promising ways to handle electrification requirements in remote regions (e.g. island) [3].

2.2. HRES optimization

Optimal design. In order to obtain electricity from an HRES reliably and economically, an optimized sizing method is necessary. To this aim, Gupta et al. [4] present the analysis and design of a mixed-integer linear mathematical programming model to determine the optimal configuration and cost for an HRES. This consists of a PV array, biomass (fuelwood), biogas, small/micro-hydro, a fossil fuel generator and a battery bank. The cost function to be minimized is based on demand and potential constraints. Particularly, the optimal sizing of such systems requires detailed analysis for a given location. There are indeed various site-dependent variables such as solar radiation, wind speed and temperature that influence to the system cost [3]. This design problem has the goal to determine the power system optimal configuration and location, type and sizing of generation units installed at certain nodes, in order to meet load requirements at minimum cost. Thus, the optimal HRES configuration seeks a combination of generator types and sizes resulting in the lowest lifetime cost and/or emission. Among all possible HRES configurations that are optimally dispatched, the configuration with the lowest "Net Present Cost (NPC)" is declared as the "optimal configuration" or the "optimal design". Yang et al. [5] presented a method for the optimization of hybrid PV-Wind-battery systems which minimize the "Levelized Cost of Energy (LCE)". The optimization is carried out by changing component combinations: number and orientation of PV modules, rated power and tower height of wind turbine, capacity of the battery bank. Summarizing, there are two possibile objective functions to be minimized for optimal design.

- Net Present Cost (NPC): investment costs plus the discounted present values of all future costs during the system lifetime;
- Levelized Cost of Energy (LCE): total cost of the entire HRES divided by the energy self produced.

Additionally, reliability restrictions are usually included, evaluating the objective function by means of a probability parameter [6].

Operational optimization. The HRES studied in this work has not to be sized because device properties are already given as input data and so are the electrical loads and the thermal loads, where present. The optimization is then carried out adjusting the operating setpoints of each HRES device. The optimization system must compute the power production profile, when an electrical load has to start, if it is convenient to charge a battery or not, and so on. A wide literature on this theme exists. Barley et al. [7] face the problem of optimal dispatch strategy for HRES in remote areas. Ashari et al. [8] present dispatch strategies for the operation of a PV-diesel-battery HRES using setpoints. The number of startup for the the diesel generator is optimized in order to minimize the overall system costs. Wang et al. [9] develop energy management strategies from both the demand side and generation side. The intended goal is to satisfy the electricity demand while minimizing both the overall operating cost and environmental impact. The latter one is accounted for by indicators of equivalent cost. Day-ahead and real-time weather forecasting, demand response and model updating are also integrated using a receding horizon optimization strategy. HRES operational optimization finds also other applications as in Park et al. [10]. The authors propose an operation control of a PV-diesel HRES for a small ship considering the PV power fluctuation due to solar radiation. The control aim is to minimize the fuel consumption with the smallest battery storage capacity. Another energy management application is the one in Wang et al. [11] in which the HRES (PV-Wind-fuel cell) is used to manage the energy flows in the chlorine-alkali process using receding horizon optimization techniques. Enyard et al. [12] use a model predictive controller (MPC) to command the flow of water passing through a storage tank, the wood boiler setpoint temperature to reduce CO_2 emissions and operating cost of a boiler system. In HRES optimization, weather forecasting is also a primary task to deal with. Many works in literature are interested in proper and efficient forecasting techniques. Among the many the authors suggest [13, 14, 15] and references therein. HRES operational optimization is also relevant in the so-called "Smart Grid" research field. Samadi et al. [16] propose a novel real-time pricing algorithm for smart grid, considering the importance of energy pricing as an essential tool to develop efficient demand side management strategies. The algorithm aims to find the optimal energy consumption levels for each subscriber to the grid, maximizing the aggregate utility of all subscribers in a fair and efficient manner. Zhu et al. [17] also proposed a consumption scheduling mechanism for home area load management in smart grid, but using an integer linear programming (ILP) technique. Wu et al. [18] minimize electricity cost subject to a number of constraints, such as power balance, solar output and battery capacity. Considering demand side management, an optimal control method (open loop) is developed to schedule the HRES power flow over 24 h. MPC is then used as closed-loop method to dispatch the power flow in real-time when uncertain disturbances occur. MPC has been used also by Wei et al. [19] to operate a Wind-PV system. The authors take firstly into account short-term optimal maintenance and operation considerations. Then, long-term optimal operation with battery maintenance and time-varying electric power pricing is considered. An extensive literature survey on HRES applied to smart grid and micro-grid can be found in [20]. A framework of diverse objectives optimized to empower the micro-grid has been outlined. A review about modeling and applications of renewable energy generation and storage sources is also presented in [20].

Optimization techniques and tools. Various optimization techniques for HRES optimization have been reported in literature. The most common ones are genetic algorithm (GA) [21, 22, 23, 5], simulated annealing (SA) [24], and particle swarm optimization (PSO) [25, 26, 27]. There are also possible promising techniques for future use in HRES sizing, such as ant colony optimization (ACO) [28] or artificial immune system (AIS) algorithm [29]. Besides, many software tools are commercially available that can be helpful for real-time system integration. The most used are: "Hybrid Optimization Model for Electric Renewables (HOMER)" [30], as the most famous, "Hybrid Power System Simulation Model (HYBRID2)" [31], "improved Hybrid Optimization by Genetic Algorithms (iHOGA)" [32], and so on. Several more optimization tools are also available for hybrid systems design [6]. A detailed literature survey specifically on commercially available software for the HRES performance evaluation, can be found in [33].

Summary. As anticipated, in this work we present an optimization system able to perform an operational optimization of an HRES. In particular, we propose to optimize an already sized energy system, which means that this tool can be adapted also to pre-existing HRES. Our main novelty is a flexible and modular modeling approach, obtained by considering every device as a single unit that can generate or absorb (electrical or thermal) power, as appropriate depending on the imposed constraints and on the economical convenience, and that contributes to an overall cost. The optimization problem objective is constituted by a sum of costs, fees, and prizes due to fulfilling or not certain energy requirements. In this sense, within the usual framework of optimization and control systems, our optimization layer can be more assimilated to the concept of Dynamic Real-Time Optimization [34, 35, 36, 37, 38, 39, 40] as its result is an economically optimal sequence of setpoints spanned on a specific time horizon.

3. HRES model

3.1. Introduction

The HRES considered in this work can be composed by several "devices" belonging to four different classes: electrical generators, electrical accumulators, electrical loads and thermal configurations [41]. A general description of each class, is given in §3.2. Each device model takes a setpoint, as input variable, ranging in $[0,1]$, except for batteries where the setpoint ranges in $[-1,1]$. Any other quantity in each device model is calculated from these setpoints: for instance, in a fuel burning electrical generator, the device input is the ratio between generated power and nominal power, while fuel consumption and generated power are outputs of the device model. Many devices present some constraints to fulfill, e.g. bounds on the state of charge (SOC) for batteries, or maximum number of startups for fuel generators. Every model device gives, as calculated output, its contribution to the cost function and to the constraint vector. Let $W(i)$ denote the net electrical power supplied by HRES to the network at instant i. Let this quantity be positive if the HRES is indeed selling electricity to network or negative if the HRES is buying electricity from the network. At each instant $i \in \{1,\dots,N\}$ the exchange of power with the network is expressed by:

$$W(i) = \sum_{k\in\mathscr{K}} G(k,i) - \sum_{m\in\mathscr{M}} C(m,i) + \sum_{b\in\mathscr{B}} A(b,i), \qquad (1)$$

in which:

- $G(k,i)$ is the power generated by the $k-$th generator;
- $C(m,i)$ is the power absorbed by the $m-$th electrical load;
- $A(b,i)$ is the power released by the $b-$th accumulator, negative when the accumulator is charged.

Note that $\mathscr{K}$ is the set of all devices that can generate electricity, $\mathscr{M}$ is the set of electrical loads, $\mathscr{B}$ is the set of batteries.

The typical time horizon considered in this work is 24 hours, divided into $N = 96$ time steps, each of length $\tau = 0.25$ h. The horizon N, and also time step length τ, can be changed according to specific requirements. Typically, the optimization tool is run one day ahead using forecasts of weather conditions, load demands, power exchange declared profile, etc. Results of this optimization run are then used as setpoints for the HRES control system. However, it can also be re-run during the current day to re-optimize the HRES operation in response to changes in weather parameters, loads, etc., or in response to a demand from the Dispatching Service Market of power exchange profile variation. In this case the horizon can be shrunk accordingly to cover the remaining portion of the current day. The computational efficiency of the developed tool is that, in principle, for typical HRES it can be re-run at each time step similarly to an MPC.

3.2. Devices models

Electrical generators. Three different electrical generators are considered: photovoltaic (PV), wind turbine (WT) and fuel burning generators. All generators take a vector of N setpoints meant as the ratio between the actual electrical power and the device nominal power over the time horizon. Another characteristic of these devices is the fuel that enters them: for PV or WT, the fuel is obviously priceless being sun and wind respectively. All generators have nameplate data as input parameters, and for the fuel burning ones, also the kind of fuel has to be specified, e.g. biomass, diesel, natural gas. Generated power [kWe] and the fuel rate [kg h^{-1}] profile over the time horizon are the outputs calculated for all generators. The general formula for the electrical power production of the k-th generator is:

$$G(k,i) = \phi_1(k,i)\alpha(k,i) \tag{2}$$

where ϕ_1 formulation depends on the specific generator. For fuel burning generators the correlation for fuel consumption is:

$$F(k,i) = \frac{G(k,i)}{LHV(k)\eta_e(k,i)} \tag{3}$$

where LHV is the lower heating value and η_e is the electrical efficiency.

Electrical accumulators. Two electrical accumulator models are defined, which differ in the rate of charging/discharging: the "BMS" one slows down the charge/discharge rate once a certain State-Of-Charge (SOC) value is reached, whereas such a limitation is not present in the conventional accumulator model. Accumulators take a vector of N setpoints meant as the ratio between the actual electrical power, accumulated or released, and the device nominal power deducted by a calculated efficiency. Other input parameters are nameplate data, e.g. charging and discharging efficiencies, SOC bounds and the initial SOC value. Released/absorbed power [kWe] and the SOC [%] profile over the selected time horizon are the two main outputs of these devices.
The general formula for the electrical power production/absorption of the b-th accumulator is:

$$A(b,i) = \psi_1(b,i)\eta_1(b,i)\beta(b,i) \tag{4}$$

where ψ_1 formulation depends on the battery nominal power and η_1 is the accumulator power exchange efficiency. The SOC profile correlation is then:

$$SOC(b,i) = SOC(b,i-1) + \psi_2(b) + \psi_3(b)\eta_2(b,i)\beta(b,i) \tag{5}$$

where ψ_2 and ψ_3 counts for internal electrical effects and η_2 is the accumular storage efficiency.

Electrical loads. Three different electrical load models are here considered: L_1, L_2 and L_3 loads. L_1 types are used to represent all mandatory, non adjustable electrical consumptions. L_2 types are used to represent electrical consumption cycles which need to be completed (one or more times) at no specific time over the time horizon. L_3 types, instead, represent loads normally on, that can be shut down for a limited amount of time without compromising the related process operation. Setpoints for the loads are here meant as the starting and ending times of each cycle: obviously L_1 loads do not have any setpoint as they are fixed. The electrical absorbed power [kWe] profile over the time horizon is its only output calculated.
The general formula for the electrical power consumption of the m-th electric load is:

$$C(m,i) = f_L(\gamma(m,i),i) \tag{6}$$

where f_L depends on the load type and γ is the setpoint for the time-varying loads.

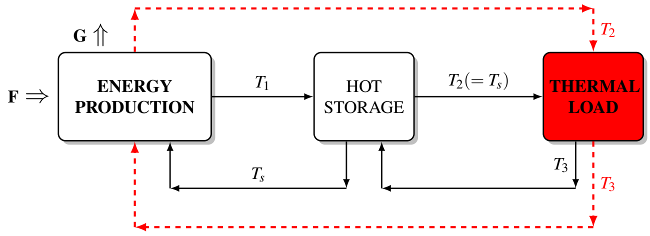

Figure 1: General block diagram of HOT thermal configurations. The black continuous lines represent the path followed in most of the configuration, while the red dotted lines indicate a direct exchange between the energy production device and the thermal load.

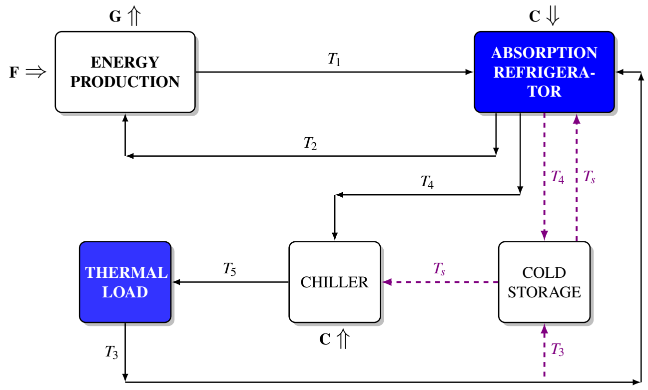

Figure 2: General block diagram of COLD thermal configurations. The black continuous lines represent the path followed in most of the configuration, while the purple dotted lines indicate the paths followed in the presence of a cold storage tank.

Thermal Configurations. The thermal configurations are divided into two categories depending on the purpose of heat transfer, i.e. whether heat is supplied to or removed from the thermal utilizer.

As depicted in Figure 1, thermal configurations denoted as "HOT" supply heat by means of a hot medium stream, usually water at $80 \div 90$°C. This material stream enters the thermal load at temperature T_2 and leaves it at temperature T_3, with $T_3 < T_2$. Many different configurations are possible, with or without intermediate hot storage tanks. In Figure 1 the block named "Energy Production" represents one or more devices that use a "stream" **F**, fuel or electricity, to produce the heated material stream at temperature T_1 sent to a hot storage tank, or at temperature T_2 in case of direct exchange with the thermal load. In some configurations, electrical power **G** can be produced, usually through a combined heat and power system (CHP), and utilized in the HRES or sold to the grid. It can be noticed that, in case of multiple energy producers, an input setpoint is required for everyone of them.

As depicted in Figure 2, thermal configurations denoted as "COLD" remove heat at low temperature (e.g., $10 \div 12$°C using chilled water) in order to satisfy a generic thermal load. A material stream at temperature T_5 is sent to the thermal load, and leaves it at temperature T_3, with $T_3 > T_5$. Also in this case different formulations are possible, with or without intermediate cold storage tanks. In Figure 2 there is still a block named "Energy Production" representing a device that uses a material fuel stream **F** to produce the heated stream at temperature T_1 that drives an absorption refrigerator [42]. As for HOT configurations, electrical power **G** can be produced, and in case of multiple energy producers, an input setpoint has to be defined for each of them.

Each, HOT or COLD, thermal configuration takes the thermal load profile requirements as parameters and gives the thermal and generated/absorbed electrical power profiles over the time horizon as outputs. In addition, all nameplate data and fuel type used must be specified.

Due to the complexity of these devices, a single general mathematical formula cannot be given. The specific formulation for a particular thermal configuration can be seen in the example in § 6.2.

3.3. Objective function

The objective function to be minimized, denoted by f, has the following structure:

$$f = \text{Costs} - \text{Revenues} + \text{Penalties} \tag{7}$$

in which:

- Costs are associated to electricity bought from the network and fuel consumption.
- Revenues are associated to electricity sold to the network and to incentives (e.g., for power generation using renewable sources).
- Penalties are associated to not respecting a power generation profile.

This objective function can be slightly different depending on the specific tariff regime. In this work four different tariff regimes may apply to an HRES: they represent the four most common energy policies currently available in Italy as established by law. As an example, one of the four tariff is explained and analyzed below to let the reader understand the objective function construction rationale.

3.4. Example of tariff regime

In this tariff regime, the cost function can be expressed by:

$$f = \sum_{i=1}^{N} [f_W(i) + f_F(i) - f_I(i) + f_D(i)] \tag{8}$$

in which:

- $f_W(i)$ is the positive or negative cost associated to exchange of electricity, during the $i-$th time step.
- $f_F(i)$ is the positive cost associated to fuel consumption, during the $i-$th time step.
- $f_I(i)$ is the positive incentive awarded, during the $i-$th time step, for power generation by means of renewable sources or High Efficiency Co-Generation (HECG) systems.
- $f_D(i)$ is the positive cost associated to penalties for missed production and/or the negative cost associated to successful responses to requests from the Dispatching Service Market (DSM), during the $i-$th time step.

Cost of electrical energy exchange. The cost of selling/buying electrical energy to/from the network (actual exchange) is evaluated as follows:

$$f_W(i) = c_W(i)W(i), \text{ with } c_W(i) = \begin{cases} -p_S(i)\tau & \text{if } W(i) \geq 0 \\ -p_B(i)\tau & \text{if } W(i) < 0 \end{cases} \tag{9}$$

where $p_S(i)$, $p_B(i)$ are the positive selling and buying electricity prices [€/kWh] at each time step over the time horizon, and τ is the time step length [h]. Notice that $c_W(i) \leq 0$, $\forall i$. Thus, $f_W(i) \geq 0$ when $W(i) < 0$ i.e. when the HRES buys electricity from the network, and $f_W(i) \leq 0$ when $W(i) \geq 0$ i.e. when the HRES sells electricity to the network.

Cost of fuel consumption. The fuel consumption cost for electrical generators, HOT and COLD configurations is expressed as:

$$f_F(i) = \sum_{k \in \mathcal{K}} c_F(k)\tau F(k,i) \tag{10}$$

where $F(k,i)$ is the fuel rate [kg/h] (at the i-th time step and for the k-th generator) and $c_F(k)$ is its unit price [€/kg].

Incentives for renewable generation and HECG systems. The incentives for generation from renewable sources apply when the HRES is composed by renewable generators of same type, i.e. only PV or WT or biomass burning generators (BM), and electrical loads. The incentives for HECG systems ("White Certificates", WC, and "Excise Tax reduction for HECG fueled with Natural Gas", NG) can apply together, but all other incentives are lost. We can write all incentives at the i-th time step as the following sum:

$$f_I(i) = f_{PV}(i) + f_{WT}(i) + f_{BM}(i) + f_{WC}(i) + f_{NG}(i) \tag{11}$$

The first three terms represent the renewable contributions, while both WC and NG are related to fuel burning generators with specific requirements on the efficiency and on the fuel burned, respectively. Except for specific waived cases, for a given HRES, according to the Italian energy policy, if the fourth and/or fifth term are nonzero, then the first three terms are zero. On the other hand only one of the first three terms can be nonzero, and in such case the fourth and fifth term are also zero.

Incentives and penalties of Dispatching Service Market (DSM). Penalties are charged when the declared exchange of electricity is not respected, within a predefined tolerance. For each time step, we define $P_W(i)$ the penalty to pay for exchanging $W(i)$

different from $\bar{W}(i)$. Furthermore, incentives are awarded if the HRES responds successfully to a DSM request of variation. Such a request is defined in terms of a vector of N components $DSM(i)$ representing a positive or negative variation from the declared power exchange $\bar{W}(i)$. For each time step, we define $I_D(i)$ as the incentive awarded.

It is useful to define the combined cost:

$$f_D(i) = P_W(i) - I_D(i) \tag{12}$$

So, the term $f_D(i)$ can be written as follows:

$$f_D(i) = c_{W,D}(i)W(i) + \bar{f}_W(i) \tag{13}$$

in which $c_{W,D}(i)$ and $\bar{f}_W(i)$ are suitably defined depending on the sign and the value of $(W(i) - \bar{W}(i))$.

Tariff summary. Collecting all terms together, the objective function can be finally written as follows:

$$f_T = \sum_{i=1}^{N} f(i) \tag{14}$$

in which

$$f(i) = \bar{f}_W(i) + c_{W,T}(i)W(i) - \sum_{k\in\mathscr{K}} c_G(k)G(k,i) - \sum_{k\in\mathscr{K}_{HECG}} c_{Q,WC}(k)Q_{CHP}(k,i) + \sum_{k\in\mathscr{K}} c_F(k)F(k,i) \tag{15}$$

with

$$c_{W,T}(i) = c_W(i) + c_{W,D}(i) \tag{16}$$

Few terms in (15) need to be explained: $c_G(k)$ is the coefficient associated to the electrical power generation [€/kW]; $c_{Q,WC}(k)$ is the coefficient associated to the heat power $Q_{CHP}(k,i)$ [kW] generated by the CHP which earns the WC incentive [€/kW].

Finally, for every device, it is possibile to calculate its associated cost $\hat{f}(i)$, selecting the specific terms of (15). For instance, for electrical loads it will be only $\bar{f}_W(i) + c_{W,T}(i)W(i)$. However, it is important to point out that despite this separability of the cost function into specific contributions of each HRES device, from (9), (13) and (16) it follows that the coefficients $c_{W,T}(i)$ depend on the overall power exchange $W(i)$, thus coupling the cost function among all devices.

4. Mathematical problem formulation

The general formulation of the optimization problem to be solved can be written as follows. Let $x \in \mathbb{R}^{n_x}$ denote the stacked vector of all device setpoints, and let $x_{\min} \in \mathbb{R}^{n_x}$ and $x_{\max} \in \mathbb{R}^{n_x}$ denote the associated bound constraints, i.e.

$$x = \begin{bmatrix} \beta(1) \\ \vdots \\ \beta(\mathscr{N}_b) \\ \gamma(1) \\ \vdots \\ \gamma(\mathscr{N}_m) \\ \alpha(1) \\ \vdots \\ \alpha(\mathscr{N}_k) \end{bmatrix}, \quad x_{\min} = \begin{bmatrix} -\mathbf{1} \\ \vdots \\ -\mathbf{1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad x_{\max} = \begin{bmatrix} \mathbf{1} \\ \vdots \\ \mathbf{1} \\ \mathbf{1} \\ \vdots \\ \mathbf{1} \\ \mathbf{1} \\ \vdots \\ \mathbf{1} \end{bmatrix},$$

in which $\mathscr{N}_b$ is the number of batteries, $\mathscr{N}_m$ is the number of electrical loads, $\mathscr{N}_k$ is the number of generators plus thermal configurations, $\mathbf{1}$ and $\mathbf{0}$ are vectors of suitable dimensions filled with ones and zeros, respectively. Thus $\beta(b)$ is the vector of setpoints for the b-th accumulator, $\gamma(m)$ is the vector of setpoints for the m-th electrical load, and $\alpha(k)$ is the vector of setpoints for the k-th electrical generator. We notice that for L_1 loads, the setpoint and corresponding bound vectors are empty because this device does not have any decision variable, but it affects the cost function because the sign of the cost of exchanged electricity $c_W(i)$ depends on $W(i)$. As anticipated in §3.1, a number of devices (e.g., accumulators or thermal configurations) have process constraints in addition to bound constraints on their setpoints. Let $c_\alpha(k)$ be the (possibly empty) constraint vector for the $k-$th generator and thermal configuration, $c_\beta(b)$ the constraint vector for the $b-$th battery, $c_\gamma(m)$ the (possibly empty) constraint vector for the $m-$th electrical load. These process constraint vectors can be stacked together obtaining a single vector of constraints:

$$c(x) = \begin{bmatrix} c_\beta(1) \\ \vdots \\ c_\beta(\mathscr{N}_b) \\ c_\gamma(1) \\ \vdots \\ c_\gamma(\mathscr{N}_m) \\ c_\alpha(1) \\ \vdots \\ c_\alpha(\mathscr{N}_k) \end{bmatrix} \leq \mathbf{0}$$

Overall, we denote by n_{in} the dimension of $c(x)$, i.e. $c(x) \in \mathbb{R}^{n_{in}}$.

The optimization problem, in specific conditions, is also required to satisfy a vector of equality constraints on the overall district power exchanged at each time step, i.e.:

$$W(i) \triangleq \sum_{k=1}^{\mathscr{N}_k} W(k,i) + \sum_{b=1}^{\mathscr{N}_b} W(b,i) + \sum_{m=1}^{\mathscr{N}_m} W(m,i) = \bar{W}(i), \quad i = 1,\dots,N \tag{17}$$

in which, as anticipated in §3.1, $\bar{W}(i)$ is the desired value of exchanged power at each time step. For a stand-alone HRES, clearly $\bar{W}(i) = 0$ for all $i \in \{1,\dots,N\}$. On the other hand, for a grid-connected HRES, $\bar{W}(i)$ represents a power exchange profile that the HRES *must* exchange with the network. Constraint (17) is expressed in the following form:

$$c_{eq}(x) = \begin{bmatrix} W(1) - \bar{W}(1) \\ \vdots \\ W(N) - \bar{W}(N) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{18}$$

in which $c_{eq}(x) \in \mathbb{R}^N$. In some HRES, it is tolerable to satisfy a relaxed version of (18), as follows:

$$-\mathbf{1}\varepsilon_1 \leq c_{eq}(x) \leq \mathbf{1}\varepsilon_1 \tag{19}$$

with $\varepsilon_1 > 0$. This case falls back to the situation where only inequality constraints are present, with the following redefinition:

$$c(x) \leftarrow \begin{bmatrix} c(x) \\ c_{eq}(x) - \mathbf{1}\varepsilon_1 \\ -c_{eq}(x) - \mathbf{1}\varepsilon_1 \end{bmatrix} \tag{20}$$

Finally, as better explained in §3.3, the objective function can be expressed as the sum of the partial objective functions of all devices, i.e.

$$f(x) = \sum_{i=1}^{N} \left(\sum_{k=1}^{\mathcal{N}_k} \hat{f}(k,i) + \sum_{b=1}^{\mathcal{N}_b} \hat{f}(b,i) + \sum_{m=1}^{\mathcal{N}_m} \hat{f}(m,i) \right),$$

in which we notice that the inner sums define the overall district cost of each time instant $i \in \{1, \ldots, N\}$, i.e.

$$f(i) = \sum_{k=1}^{\mathcal{N}_k} \hat{f}(k,i) + \sum_{b=1}^{\mathcal{N}_b} \hat{f}(b,i) + \sum_{m=1}^{\mathcal{N}_m} \hat{f}(m,i),$$

and the outer sum calculates the overall (daily) cost.

Thus, the nonlinear program to be solved is in the form:

$$\min_{x} \; f(x), \tag{21a}$$

subject to

$$x_{\min} \leq x \leq x_{\max} \tag{21b}$$
$$c(x) \leq \mathbf{0} \tag{21c}$$
$$c_{eq}(x) = \mathbf{0} \tag{21d}$$

in which $x \in \mathbb{R}^{n_x}$, $c(x) \in \mathbb{R}^{n_{in}}$, $c_{eq}(x) \in \mathbb{R}^N$.

5. Optimization algorithm

The main theoretical foundations of the Sequential Linear Programming (SLP) algorithm developed in this work are now discussed.

5.1. General SLP formulation

There are various reasons why it was decided to implement an SLP solver for this kind of problem. Non-linearity of most of the model devices and objective function suggest us to solve a general NLP as in (21). In addition, several optimization variables are in principle binary since devices can be on or off. Moreover, we aimed at developing a tool able to handle quite large HRES, leading to mixed-integer nonlinear programming (MINLP) problems in sever hundreds/thousands of variables, which cannot be efficiently tackled by off-the-shelf solvers. Therefore, it has been decided to apply a smoothed replacement for the integer variables (as for batteries and generators switch) in order to avoid an MINLP approach. Sequential Quadratic Programming (SQP) methods require second order information (Hessian matrix), and most of these utilize approximated information (i.e. Broyden matrix) while in the SLP method this is not necessary. Furthermore, there are many reliable, large-scale, open-source LP solvers, while much less QP solvers are available and overall they are less efficient. In the end, since each device setpoint does not depend, in terms of local feasibility, on other device setpoint makes the SLP approach the best choice for this particular problem structure. In particular, when no global power profile constraints exist, one could parallelize the local LPs and solve them separately for each device of the HRES.

The considered approach falls in the class of *nonsmooth penalty methods* [43, Sect. 17.2] implemented within a *trust region* framework [43, Chp. 4]. Starting from a feasible initial guess is not required, as well as feasibility of the nonlinear constraints (21c) (and of (21d)), is not necessarily maintained at each iteration. Then, if the feasible region is nonempty, the algorithm recovers a feasible point and then converges to a local minimum or, otherwise, it reports that the problem is infeasible. The following nonsmooth cost function, associated with the original nonlinear program (21) is defined:

$$\Phi(x;\mu) = f(x) + \mu \sum_i |c_{eq,i}(x)| + \mu \sum_i [c_i(x)]^+ \tag{22}$$

in which $[y]^+ = \max\{0, y\}$ for each $y \in \mathbb{R}$, and $\mu > 0$. At each iteration, for a given μ, we make an attempt to solve the following nonsmooth NLP optimization problem, with bound constraints only:

$$\min_{x} \Phi(x;\mu) \tag{23a}$$

subject to

$$x_{\min} \leq x \leq x_{\max} \tag{23b}$$

The penalty parameter μ is chosen large and increased if necessary to promote feasible iterates.

The following smooth replacement for $\Phi(x;\mu)$ in (23) is then considered:

$$\tilde{\Phi}(\xi;\mu) = f(x) + \mu \sum_i \bar{s}_i + \mu \sum_i \underline{s}_i + \mu \sum_i s_i \tag{24a}$$

subject to:

$$c(x) \leq s \tag{24b}$$

$$c_{eq}(x) = \overline{s} - \underline{s} \tag{24c}$$

$$s, \overline{s}, \underline{s} \geq \mathbf{0} \tag{24d}$$

in which

$$\xi = \begin{bmatrix} x \\ s \\ \overline{s} \\ \underline{s} \end{bmatrix}, \tag{25}$$

is the augmented decision variable. Thus, problem (24) is the one solved in the algorithm with an SLP procedure using a trust region method. In preparation to the algorithm, the following definitions are made:

$$\xi_{\min} = \begin{bmatrix} x_{\min} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad \xi_{\max} = \begin{bmatrix} x_{\max} \\ \boldsymbol{\infty} \\ \boldsymbol{\infty} \\ \boldsymbol{\infty} \end{bmatrix}, \quad \Psi(\xi) = c(x) - s,$$

$$\Gamma(\xi) = c_{eq}(x) - \overline{s} + \underline{s}, \quad \nabla\tilde{\Phi}(\xi;\mu) = \begin{bmatrix} \nabla f(x) \\ \mathbf{1}\mu \\ \mathbf{1}\mu \\ \mathbf{1}\mu \end{bmatrix},$$

$$\nabla\Psi(\xi) = \begin{bmatrix} \nabla c(x) \\ -\boldsymbol{I} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad \nabla\Gamma(\xi) = \begin{bmatrix} \nabla c_{eq}(x) \\ \mathbf{0} \\ -\boldsymbol{I} \\ \boldsymbol{I} \end{bmatrix} \tag{26}$$

in which $\boldsymbol{\infty}$ is a vector of "infinity", $\mathbf{0}$ is vector/matrix of zeros, and $\boldsymbol{I}$ is the identity matrix, each of suitable dimensions. Let x_j denote the vector x at the $j-$th iteration of the SLP algorithm described next. Likewise, let ξ_j denote the augmented vector ξ at the $j-$th iteration. Finally, let $\Delta_j > 0$ denote the trust region radius at the current $j-$th iteration of the SLP algorithm. The LP subproblem to be solved at the j-th iteration is the following:

$$\min_{p} \quad \nabla\tilde{\Phi}(\xi_j;\mu_j)^T p \tag{27a}$$

subject to:

$$\tilde{\Psi}(\xi_j) + \nabla\tilde{\Psi}(\xi_j)^T p \leq \mathbf{0} \tag{27b}$$

$$\xi_{\min} \leq \xi_j + p \leq \xi_{\max} \tag{27c}$$

$$-\mathbf{1}\Delta_j \leq p_j \leq \mathbf{1}\Delta_j \tag{27d}$$

in which $p = \begin{bmatrix} p_x^T & p_s^T & \overline{p}_s^T & \underline{p}_s^T \end{bmatrix}^T$ and

$$\tilde{\Psi}(\xi) = \begin{bmatrix} \Psi(\xi) \\ \Gamma(\xi) \\ -\Gamma(\xi) \end{bmatrix}, \qquad \nabla\tilde{\Psi}(\xi) = \begin{bmatrix} \nabla\Psi(\xi) \\ \nabla\Gamma(\xi) \\ -\nabla\Gamma(\xi) \end{bmatrix}.$$

To better clarify the algorithm structure, a block diagram is also presented in Fig. 3. Details of this scheme are given next, distinguishing between two variants: the basic algorithm uses a uniform trust region on all components, whereas the second one adopts a component based trust region.

5.2. SLP method 1 (common trust region)

As anticipated, the HRES optimization tool utilizes an SLP approach equipped with a trust region. The SLP algorithm is described in Algorithm 1, in which default parameters are: $\varepsilon = 10^{-6}$, $\varepsilon_f = 10^{-2}$, $\rho_{bad} = 0.10$ and $\rho_{good} = 0.75$.

The main core of the algorithm is the LP in (27), solved in Line 5 obtaining a candidate step p^*. Its norm is confronted with the parameter ε for a local solution test (Line 6). If no local solution is found, a feasibility check of the new candidate iterate $\xi_j + p^*$ is made (Line 9). If this check fails the trust region radius is reduced and the step rejected (Line 9). Otherwise, the step is finally accepted or rejected on the basis of the ratio between the actual reduction of $\Phi(\cdot)$ and the reduction of its smoother counterpart $\tilde{\Phi}(\cdot)$, named ρ_j (Line 10). If this parameter is greater than a default value η, then the variable ξ_{j+1} is updated with $\xi_j + p^*$ (Line 12), otherwise p^* is rejected (Line 14); this means that the step is feasible but not good enough to be applied. The parameter ρ_j value plays a final role in the trust region evolution (Lines 15–21): if ρ_j is large it means that we are confident about greater improvements and we can enlarge the trust region to let the LP subproblem take larger steps. In the opposite case, when ρ_j is too small, even if the current iteration is feasible, the next one could not be, so we shrink the trust region in order to better guarantee a feasible LP problem at the next iteration. The new slack iterates are always redefined as the actual new constraint violations (Line 22), while the penalty parameter μ is increased (most often strictly) if the current iterate is still infeasible (Line 24). Once feasibility is recovered, μ is not further increased to prevent numerical ill conditioning (Line 25).

Further comments to Algorithm 1 are useful. Line 4 finds a step from the current augmented decision variable iterate ξ_j towards the minimization of problem (27) with variables x limited by the trust region of size Δ_j. On the other hand, the step for slack variables $(s, \overline{s}, \underline{s})$ is not limited by a trust region, because these variables enter linearly in both the cost function and the constraints. The check of Line 9 is performed to see if the expected slacked constraints at the next iterate are satisfied or not. If these constraints do not hold, the behavior of the constraint functions is too nonlinear and the trust region (of the x variables) should be reduced. In addition, if the step is rejected, the trust region should not be reduced if it was already reduced by Line 9. When the step is accepted with large ρ_j, the trust region radius is enlarged to a value that is no greater than the initial ρ value. At the end, Line 27 performs the final feasibility check for the found local solution. If constraints are not satisfied, the considered NLP is reported to be infeasible.

5.3. SLP method 2 (component based trust region)

In Line 9 of Algorithm 1, when predicted constraints are violated, i.e. $\max\tilde{\Psi}(\xi_j + p^*) > \varepsilon_f$, the trust region size is reduced uniformly for all components of vector x. However, this is a conservative approach because the violated constraints may be affected by only a subset of components of x. This is particularly true for those systems in which setpoints and process constraints are separated for each device. For instance: the

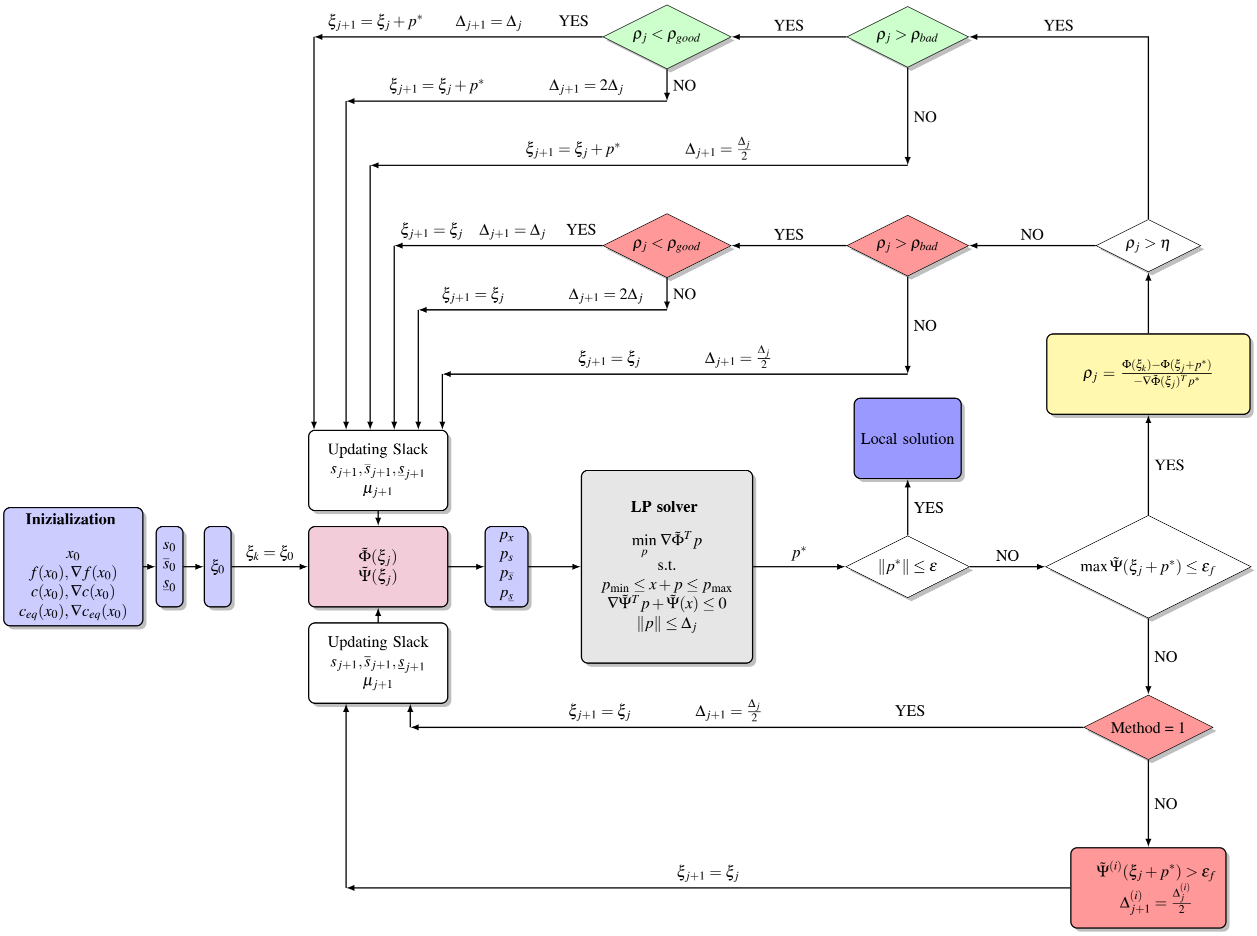

Figure 3: Block diagram of the SLP algorithm with trust region.

accumulator SOC constraint is only affected by accumulator setpoints, so it is not necessary to shrink the trust region for setpoints of other devices to prevent its violation. From this evidence, a novelty is proposed on the standard SLP 1 to improve its behavior. In particular the trust region choice has been reformulated in order to make this new algorithm variant more efficient. In this variant, each component of x has is its own trust region radius, i.e. Δ_j is a vector of length n_x. The trust region constraint imposed at the $j-$th iteration is therefore:

$$-\Delta_j \leq p_x \leq \Delta_j \tag{28}$$

For given inequality and equality constraint vectors $\Psi(\xi) = c(x) - s$ and $\Gamma(\xi) = c_{eq}(x) - \bar{s} + \underline{s}$, and a tolerance $\varepsilon_f > 0$, the following definitions are considered:

$$\begin{aligned}
&\mathscr{I}(\xi;\varepsilon_f) = \{i \in N_x \mid \exists j \in N_{in} \text{ such that:}\\
&\qquad c_j(x) - s_j > \varepsilon_f \text{ and } \left|\frac{\partial c_j(x)}{\partial x_i}\right| > 0\Big\},\\
&\mathscr{E}(\xi;\varepsilon_f) = \Big\{i \in N_x \mid \exists j \in N_{eq} \text{ such that:}\\
&\qquad |c_{eq,j}(x) - \bar{s}_j + \underline{s}_j| > \varepsilon_f \text{ and } \left|\frac{\partial c_{eq,j}(x)}{\partial x_i}\right| > 0\Big\}
\end{aligned} \tag{29}$$

in which $N_x = \{1, \ldots, n_x\}$, $N_{in} = \{1, \ldots, n_{in}\}$, and $N_{eq} = \{1, \ldots, N\}$. It has to be observed that $\mathscr{I}(\xi;\varepsilon_f)$ contains the indices of the components of x which affect inequality constraints that are violated beyond a tolerance ε_f, whereas $\mathscr{E}(\xi;\varepsilon_f)$ contains only the indices of the components of x which affect equality constraints that are violated beyond a tolerance ε_f. In particular, as shown in Fig. 3, the difference from the Algorithm 1 is just in the Line 9. Algorithm 2 reports only the changed lines.

Considerations outlined for SLP 1 hold also for SLP 2, except that the trust region reduction that occurs in Line 9 (of either algorithm) is performed in SLP 2 only for those components of x that affect the violated constraints. In this way, variables that do not affect violated constraints do not experience a shrink of their trust region, and can take possibly large steps to improve the algorithm convergence towards a local solution.

6. Applications

A brief explanation about the software implementation is now provided. Then, a case study and a discussion about the results obtained are reported.

Algorithm 1 Infeasible SLP algorithm with common trust region (SLP 1)

1: Choose: $\mu_0 > 0$, $\mu_{\max} > 0$, $0 \le \eta \le \rho_{bad}$, $0 < \sigma < 1$, and x_0 s.t. $x_{\min} \le x_0 \le x_{\max}$.
2: Compute: $\Phi(x_0)$, $s_0 = [c(x_0)]^+$, $\bar{s}_0 = [c_{eq}(x_0)]^+$, $\underline{s}_0 = [-c_{eq}(x_0)]^+$. Define: $\xi_0 = \begin{bmatrix} x_0^T & s_0^T & \bar{s}_0^T & \underline{s}_0^T \end{bmatrix}^T$. Set: $j = 0$.
3: **while** $j \le j_{max}$ **do**
4: Evaluate $\nabla\tilde{\Phi}(\xi_j;\mu_j)$, $\nabla\Psi(\xi_j)$, $\nabla\Gamma(\xi_j)$ from (26).
5: Solve LP problem (27) obtaining a candidate step $p^* = \begin{bmatrix} (p_x^*)^T & (p_s^*)^T & (\bar{p}_s^*)^T & (\underline{p}_s^*)^T \end{bmatrix}^T$, and $\lambda_{\max}$ largest Lagrange multiplier.
6: **if** $\|p^*\|_\infty \le \varepsilon$ **then**
7: $\xi^* = \begin{bmatrix} (x^*)^T & (s^*)^T & (\bar{s}^*)^T & (\underline{s}^*)^T \end{bmatrix}^T$ is a local solution to the problem (23). Go to Line 27.
8: **if** $\max\tilde{\Psi}(\xi_j + p^*) > \varepsilon_f$ **then**
9: Reject the step: $x_{j+1} = x_j$. Shrink the trust region: $\Delta_{j+1} = \frac{1}{2}\Delta_j$. Go to Line 22.
10: Compute the step evaluation parameter:

$$\rho_j = \frac{\Phi(x_j;\mu_j) - \Phi(x_j + p_x^*;\mu_j)}{-\nabla\tilde{\Phi}(\xi_j;\mu_j)^T p^*}$$

11: **if** $\rho_j \ge \eta$ **then**
12: Accept the step: $x_{j+1} = x_j + p_x^*$
13: **else**
14: Reject the step: $x_{j+1} = x_j$.
15: **if** $\rho_k \le \rho_{bad}$ **then**
16: Shrink the trust-region: $\Delta_{j+1} = \frac{1}{2}\Delta_j$
17: **else**
18: **if** $\rho_j \ge \rho_{good}$ **and** $\|p_x^*\|_\infty \ge 0.8\Delta_j$ **then**
19: Enlarge the trust-region: $\Delta_{j+1} = \min\{2\Delta_j, \Delta_0\}$
20: **else**
21: $\Delta_{j+1} = \Delta_j$.
22: Update: $s_{j+1} = \left[c(x_{j+1})\right]^+$, $\bar{s}_{j+1} = \left[c_{eq}(x_{j+1})\right]^+$, $\underline{s}_{j+1} = \left[-c_{eq}(x_{j+1})\right]^+$.
23: **if** $\max c(x_{j+1}) > \varepsilon_f$ **or** $\max|c_{eq}(x_{j+1})| > \varepsilon_f$ **then**
24: Update: $\mu_{j+1} = \min\{\max\{\mu_j/\sigma, \lambda_{\max}\}, \mu_{\max}\}$
25: **else**
26: $\mu_{j+1} = \mu_j$
27: Check the computed solution to NLP (23), ξ^*:
28: **if** $\max c(x^*) \le \varepsilon_f$ **and** $-\mathbf{1}\varepsilon_f \le c_{eq}(x^*) \le \mathbf{1}\varepsilon_f$ **then**
29: x^* is a local solution to (21).
30: **else**
31: NLP problem (21) appears infeasible.

Algorithm 2 Infeasible SLP algorithm with component trust region (SLP 2)

1: $\cdots$
2: $\cdots$
3: $\cdots$
4: $\cdots$
5: $\cdots$
6: $\cdots$
7: $\cdots$
8: **if** $\max\tilde{\Psi}(\xi_j + p^*) > \varepsilon_f$ **then**
9: Reject the step: $x_{j+1} = x_j$. Shrink the trust region of some components: $\Delta_{j+1,i} = \frac{1}{2}\Delta_{j,i}$ for all $i \in \mathscr{I}(\xi + p^*;\varepsilon_f) \cup \mathscr{E}(\xi + p^*;\varepsilon_f)$. Go to Line 22.
10: $\cdots$

6.1. Software implementation

The optimizer is implemented in C++ and compiled for both 32-bit and 64-bit Windows platforms using Microsoft Visual Studio Express 2012. The class diagram of the software architecture is depicted in Figure 4. The `District` class contains one or more device instances (`Device` implementations). Devices are grouped in sub-categories represented by the abstract classes: `Generator`, `Accumulator` and `Load`. A general tariff interface is defined by the abstract class `Tariff`. The district then contains only a particular tariff implementation (concrete tariff). Besides the modeling interfaces, the district itself is an implementation of an analysis interface denoted by the `NLPinterface` abstract class. It means that the district defines a nonlinear programming problem as the one in (21). All NLP constitutive functions are suitably constructed based on the devices contained in the district along with the specific tariff. One of the advantages of the proposed architecture is the freedom to add additional devices and tariffs without modifying the existing code. Only a new class should be added implementing the corresponding abstract interface.

All the device and tariff data are provided by the optimization tool user by means an Excel spreadsheet that is parsed by a district composer. Of course, plain C++ does not provide all the features to parse spreadsheets. In general additional packages have been used to provide particular services such as linear algebra computation or Excel spreadsheet manipulation. Significant effort has been made also to use free software, as detailed.

1. *Armadillo* (`http://arma.sourceforge.net/`): Armadillo is a C++ linear algebra library (matrix maths) aiming towards a good balance between speed and ease of use. The syntax (API) is deliberately similar to Matlab. The use of this package helps the software development while keeping the code highly readable.
2. *Excel Format Library* (`http://www.codeproject.com/Articles/42504/ExcelFormat-Library/`): The *Excel Format Library* processes spreadsheet Excel files in the BIFF8 XLS file format. It performs the basic operation as read/write operation but it also perform a cell format setting.

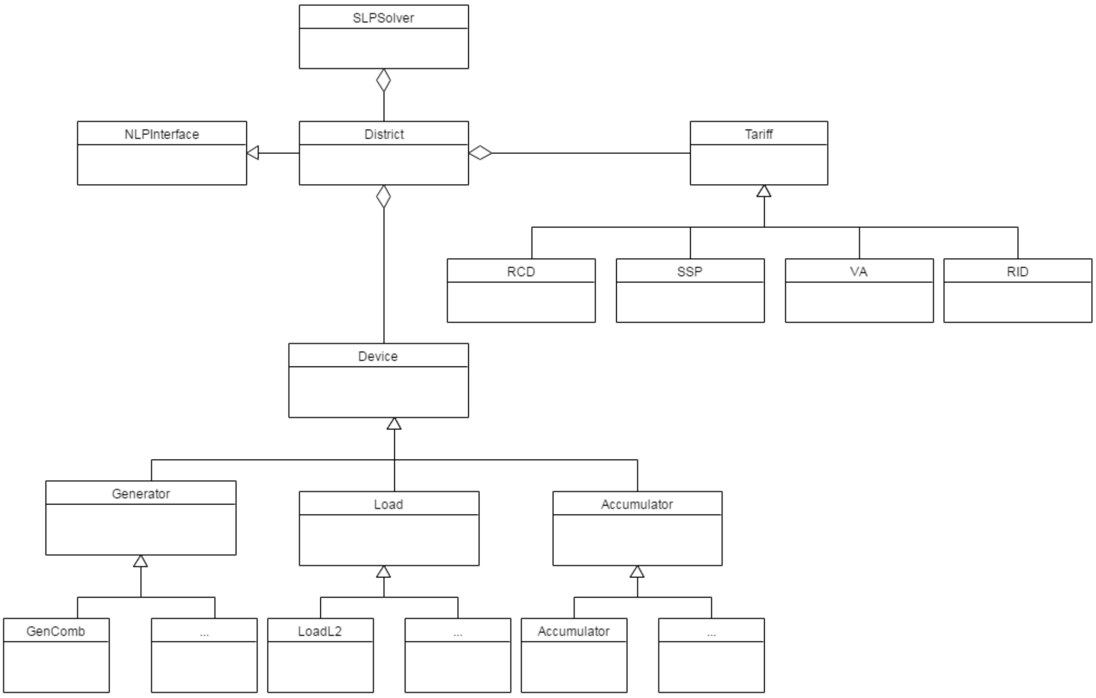

Figure 4: Optimizer Class Diagram.

3. *GLPK* (`http://www.gnu.org/software/glpk/`): As seen in §5, the optimization algorithm used to find the optimal setpoints is based on a SLP. The main core of the SLP is the linear programming problem to solve at each iteration defined in (27). In designing the software it has been convenient developing the SLP solution strategy using plain C++, and using an existing package to solve the LP problem. The GLPK (GNU Linear Programming Kit) package is intended for solving large-scale LP, mixed-integer linear programming (MILP), and other related problems. It consists in a set of routines written in ANSI C and organized in the form of a callable library. Of course an ad-hoc interface has been built between the SLP and GLPK to perform the overall optimization problem.

The Excel prototype and interface. As anticipated, the HRES definition is made in a single Excel workbook. Each workbook is composed by several sheets, one for each device and other few mandatory sheets containing general information for the HRES definition. Some environment forecast are needed, namely: wind speed, solar radiation and ambient temperature. In addition, the energy price regime can be selected and specified through all its parameters: $\bar{W}(i)$, electricity price (p_S and p_B), and all other parameters that depend on the tariff itself. All of these pieces of information have to be known in order to define the HRES model properly. In order to solve the optimization problem, the solver parameters have to be specified, as well as which one between the Algorithm 1 and Algorithm 2, is chosen.

6.2. *Case study*

A real HRES located in Tuscany is presented as case study: data for its design have been collected in a campaign of few days. Firstly the HRES modeling is detailed and then results for a specific day data are illustrated. The HRES in this example is composed by four devices: PV generator, WT generator, L_1 load and a thermal configuration (HOT 2). Its schematic representation is depicted in Figure 5, where the bold arrows represent the electric current flow.

6.2.1. *HRES description*

PV model. The PV generator model can be summarized in its power (G) calculation equation as follows:

$$G = \phi_1 \alpha \tag{30}$$

where

$$\phi_1 = \left(\frac{DNI_a}{DNI_r}\right) PN\left[1 + \gamma(T_b - T_r)\right]\eta \tag{31}$$

in which: α is the setpoint that ranges in [0,1], DNI_a is the corrected irradiation, DNI_r and T_r are the reference irradiation and the reference temperature, γ is a power correction coefficient, PN is the nominal PV array power and η is its overall efficiency. T_b is cell back temperature calculated from the cell temperature and the standard temperature difference.

WT model. The WT generator model can be summarized in its power (G) calculation equation as follows:

$$G = f_k(v)\alpha \tag{32}$$

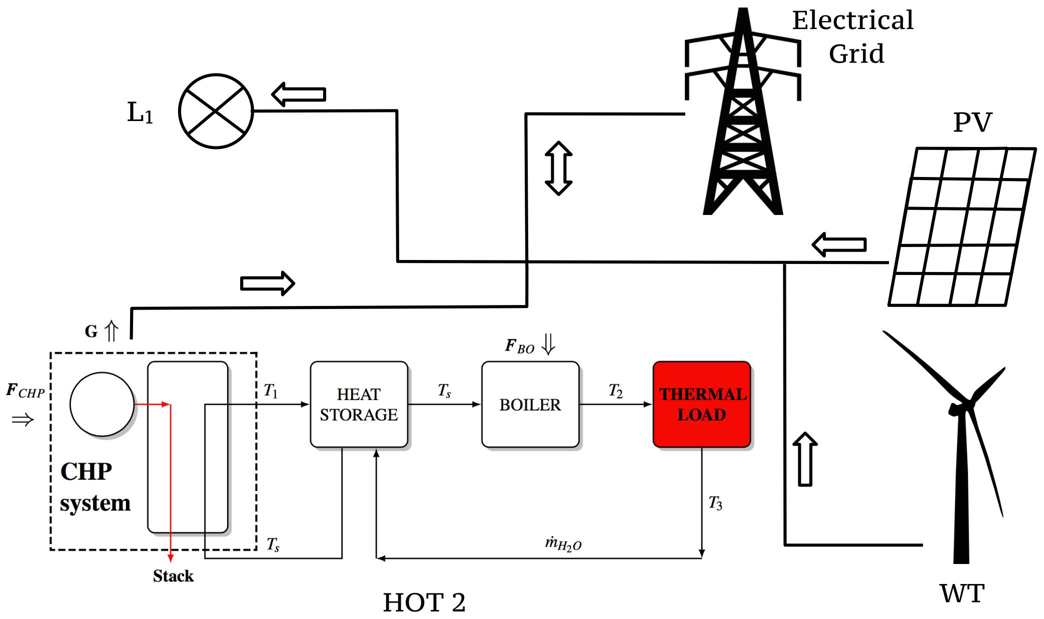

Figure 5: Test case process scheme.

where α is the setpoint that ranges in [0,1] and f_k is a function of the wind speed v that interpolates values from the WT characteristic curve.

L_1 model. The L_1 load model simply requires the electrical load profile $C(i)$ over the time horizon.

Thermal configuration HOT 2. As briefly shown in section §3, this thermal configuration has two heat generation systems, and so two decision variables need to be specified: one for the CHP (α_{CHP}) and one for the boiler, later indicated as BO, (α_{BO}). A detailed scheme of this configuration is depicted in Figure 5. The electrical power is produced only by the CHP and calculated through the following equation:

$$G = PN_{CHP}\theta_{CHP}\alpha_{CHP} \tag{33}$$

where PN_{CHP} is the nominal CHP power and θ_{CHP} represents a boolean variable ON-OFF indicating the CHP status. The thermal power, instead, is composed by two contributes, one from the CHP (Q_{CHP}) and one from the boiler (Q_{BO}), as follows:

$$Q_{CHP} = \begin{cases} PNT_{CHP}\eta_t T_{CF}\alpha_{CHP} - G & \text{if } G > 0 \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

$$Q_{BO} = PN_{BO}\theta_{BO}\alpha_{BO} \tag{35}$$

where PNT_{CHP} and PN_{BO} are the nominal thermal powers (CHP and BO respectively), η_t is the CHP total efficiency, T_{CF} is a temperature correction factor applied on the temperature of the stream from the storage, θ_{BO} represents a boolean variable ON-OFF indicating the BO status. In this way when the storage temperature (T_s) is lower than T_2, then BO will be switched on to reach the thermal requirements. On the other hand, if T_s is high enough, no additional heat by boiler is needed. T_s evolution is described by integration of the corresponding differential energy balance of the time step τ, which leads to:

$$T_s(i+1) = \left[\frac{Q_{CHP}(i)}{\dot{m}_{H_2O}C_{p,H_2O}} + T_3(i)\right](e^{\Theta} - 1) + T_s(i)e^{\Theta} \tag{36}$$

where T_3 is the thermal load outlet temperature, and

$$\Theta = \frac{\dot{m}_{H_2O}C_{p,H_2O}}{C_s}\tau \tag{37}$$

in which: C_s is the storage heat capacity, $\dot{m}_{H_2O}$ and C_{p,H_2O} are the water mass flow and specific heat, respectively. This device has also several constraints to fulfill. The first one is on the CHP maximum number of startups in order to avoid its damage, and the other one is on the thermal requirement, here expressed in terms of temperature matching: the calculated BO outlet temperature must match the thermal load inlet temperature T_2 within a tolerance ε_T.

6.2.2. *Results*

In order to assess the effective optimization benefits provided by the software to the HRES, a reference case must be identified. The selected reference case is the so called "Thermal Led" operation of the CHP, which is the standard in this HRES. This means that the CHP in this mode follows the thermal demand of the user: when the thermal demand is below the CHP minimum operational limit, heat demand is covered by the BO; furthermore the gas fired BO covers also the difference between thermal demand and maximum CHP thermal power when required. It is important to notice that, whenever the generated electrical power is lower than the required one, the HRES buys it from the grid. The starting HRES total daily cost for the selected day

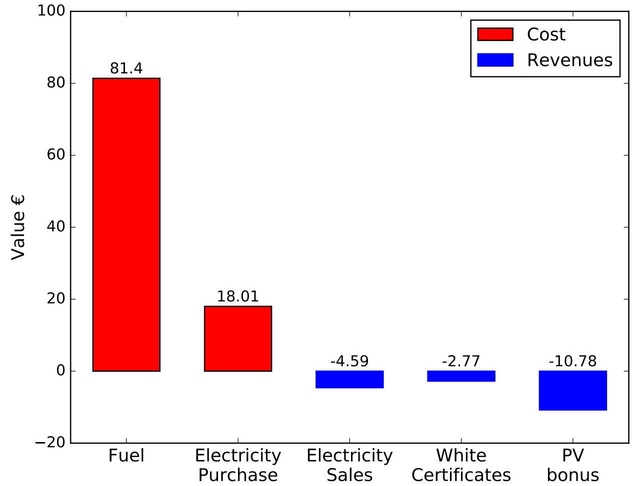

Figure 6: Contributes to the cost function.

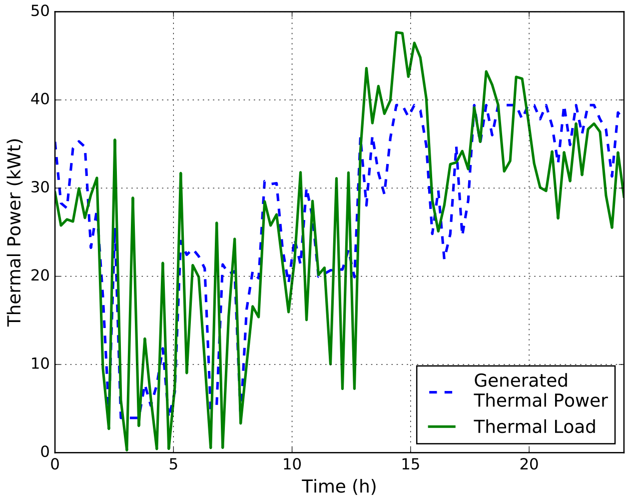

Figure 8: Thermal power profiles during the entire day.

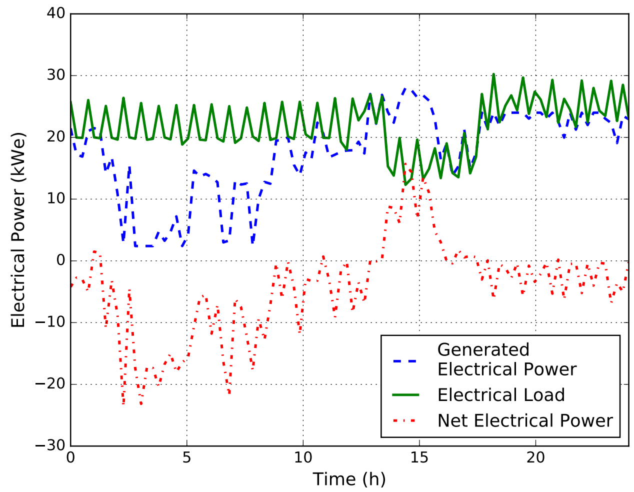

Figure 7: Electrical power profiles during the entire day.

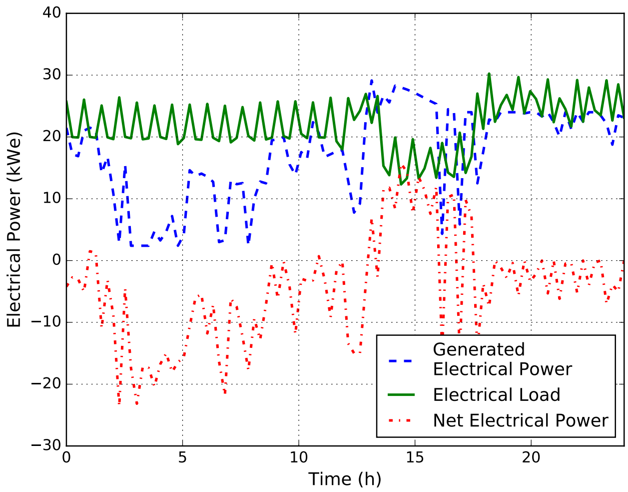

Figure 9: Electrical power profiles during the entire day. The second part of the day has changed according to the DSM request of variation.

is 89.13 €. After the optimization the economic benefits are tangible: the optimized daily cost is 81.27 €, with a cost saving of 10 % from the non-optimized case. As can be better seen in Figure 6, this HRES has benefitted incentives both from white certificates and PV presence: this system has a special energetic policy with waiver on incentives treatments. In Figures 7 and 8 the electrical and thermal power profiles are shown. From Figure 8, we observe that the algorithm tries to respect the imposed constraints on the temperature T_2 (here hidden in the thermal profile), even if it costs more tariff penalizations staying away from the declared electrical power profile, as shown in Figure 7 (in this case $\bar{W}(i) = 0 \; \forall i$).

In addition, both Algorithm 1 and Algorithm 2 have been tested, obtaining the same results. This can be explained as the problem is already feasible when entering the optimization, so the step acceptance difference has no influence here. For the same reason, no difference in processing time have been measured between the two methods in this case study.

In the end, another option of the software tool is tested. The DSM asks for a variation of the power exchange profile: in particular a negative variation of 15 kW is proposed for the time period between 12:00 and 18:00, i.e. $DSM(i) = -15$ kW $\forall \; i \in [49, 96]$. The HRES has to accept or refuse the proposed power exchange profile variation, depending on which option is more profitable. Thus, the algorithm is re-run only over the second half of the day leaving the first part unchanged. Results are shown in Figure 9, from which can be seen that the algorithm has decided to accept the DSM request of variation giving a final total day cost of 76,34 €. As can be seen from the comparison against Figure 7, the first part of the day it is the same, while the second one shows a rather accentuated modification between 12:00 and 18:00. In this case in fact, the CHP is forced to follow the market offer trying to avoid fees. As the

CHP does not have a sufficiently large nominal power, the required *DSM* cannot be achieved $\forall\ i \in [49, 96]$ but only during the central hours of the day (12:00 - 15:00, i.e. $i \in [49, 61]$) when also PV can generate electrical power. From this point, for the rest of the day (15:00 - 24:00, i.e. $i \in [61, 96]$) the CHP is still running at its maximum in order to minimize the penalty due to not achieving the required *DSM*.

7. Conclusions

In this work the problem of operation optimization of Hybrid Renewable Energy Systems (HRES) has been addressed. To this aim an HRES modeling and optimization system has been developed. Different device models, ranging from conventional, renewable, combined heat and power generators, to electrical/thermal loads and accumulators, have been considered. An operational optimization problem is formulated considering different energy policies available in Italy, and a numerical optimization algorithm has been developed. The optimization system is based on a Sequential Linear Programming (SLP) algorithm, equipped with trust region, that is able to solve a general nonlinear program: two different step acceptance possibilities have been proposed. With the modified trust region method, variables that do not affect violated constraints do not experience a shrink of their trust region and can take possibly larger steps to improve the convergence of the algorithm towards a local solution. This new proposed method gives, in most of the cases, improvements on the optimal point reached.

In the end a real case study has been analyzed. The modeling of each single device has been elaborated making it as close as possible to reality. After running the optimization algorithm, sensible improvements have been shown with a save equal to 10% for the specific case. Results show the potentialities of the developed optimization tool including the possibility of re-running the optimization for a portion of the time window in response to changes in forecasts or requests from the dispatching service market.

References

[1] V. Lazarov, G. Notton, Z. Zarkov, I. Bochev, Hybrid power systems with renewable energy sources types, structures, trends for research and development., in: Proc of International Conference ELMA, 2005, pp. 515–20.

[2] P. Bajpai, V. Dash, Hybrid renewable energy systems for power generation in stand-alone applications: A review, Renewable and Sustainable Energy Reviews 16 (5) (2012) 2926 – 2939.

[3] P. Yilmaz, M. Hakan Hocaoglu, A. E. S. Konukman, A pre-feasibility case study on integrated resource planning including renewables, Energy Policy 36 (3) (2008) 1223–1232.

[4] A. Gupta, R. Saini, M. Sharma, Modelling of hybrid energy system part i: Problem formulation and model development, Renewable Energy 36 (2) (2011) 459–465.

[5] H. Yang, Z. Wei, L. Chengzhi, Optimal design and techno-economic analysis of a hybrid solar–wind power generation system, Applied Energy 86 (2) (2009) 163–169.

[6] J. L. Bernal-Agustín, R. Dufo-López, Simulation and optimization of stand-alone hybrid renewable energy systems, Renewable and Sustainable Energy Reviews 13 (8) (2009) 2111 – 2118.

[7] C. D. Barley, C. B. Winn, Optimal dispatch strategy in remote hybrid power systems, Solar Energy 58 (4) (1996) 165–179.

[8] M. Ashari, C. Nayar, An optimum dispatch strategy using set points for a photovoltaic (pv)–diesel–battery hybrid power system, Solar Energy 66 (1) (1999) 1–9.

[9] X. Wang, A. Palazoglu, N. H. El-Farra, Operational optimization and demand response of hybrid renewable energy systems, Applied Energy 143 (2015) 324 – 335.

[10] J.-S. Park, T. Katagi, S. Yamamoto, T. Hashimoto, Operation control of photovoltaic/diesel hybrid generating system considering fluctuation of solar radiation, Solar energy materials and solar cells 67 (1) (2001) 535–542.

[11] X. Wang, H. Teichgraeber, A. Palazoglu, N. H. El-Farra, An economic receding horizon optimization approach for energy management in the chlor-alkali process with hybrid renewable energy generation, Journal of Process Control 24 (8) (2014) 1318 – 1327.

[12] J. Eynard, S. Grieu, M. Polit, Predictive control and thermal energy storage for optimizing a multi-energy district boiler, Journal of Process Control 22 (7) (2012) 1246 – 1255.

[13] D. I. Mendoza-Serrano, D. J. Chmielewski, Smart grid coordination in building {HVAC} systems: {EMPC} and the impact of forecasting, Journal of Process Control 24 (8) (2014) 1301 – 1310.

[14] V. M. Zavala, E. M. Constantinescu, T. Krause, M. Anitescu, On-line economic optimization of energy systems using weather forecast information, Journal of Process Control 19 (10) (2009) 1725 – 1736.

[15] J. Eynard, S. Grieu, M. Polit, Wavelet-based multi-resolution analysis and artificial neural networks for forecasting temperature and thermal power consumption, Engineering Applications of Artificial Intelligence 24 (3) (2011) 501 – 516.

[16] P. Samadi, A. H. Mohsenian-Rad, R. Schober, V. W. S. Wong, J. Jatskevich, Optimal real-time pricing algorithm based on utility maximization for smart grid, in: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, 2010, pp. 415–420.

[17] Z. Zhu, J. Tang, S. Lambotharan, W. H. Chin, Z. Fan, An integer linear programming based optimization for home demand-side management in smart grid, in: 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), 2012, pp. 1–5.

[18] Z. Wu, H. Tazvinga, X. Xia, Demand side management of photovoltaic-battery hybrid system, Applied Energy 148 (2015) 294 – 304.

[19] W. Qi, J. Liu, P. D. Christofides, A distributed control framework for smart grid development: Energy/water system optimal operation and electric grid integration, Journal of Process Control 21 (10) (2011) 1504 – 1516.

[20] A. H. Fathima, K. Palanisamy, Optimization in microgrids with hybrid energy systems: A review, Renewable and Sustainable Energy Reviews 45 (2015) 431 – 446.

[21] J. H. Holland, Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence., U Michigan Press, 1975.

[22] S.-M. Chen, C.-M. Huang, A new approach to generate weighted fuzzy rules using genetic algorithms for estimating null values, Expert Systems with Applications 35 (3) (2008) 905–917.

[23] O. Erdinc, M. Uzunoglu, Optimum design of hybrid renewable energy systems: Overview of different approaches, Renewable and Sustainable Energy Reviews 16 (3) (2012) 1412 – 1425.

[24] S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies, Journal of statistical physics 34 (5-6) (1984) 975–986.

[25] N. Phuangpornpitak, W. Prommee, S. Tia, W. Phuangpornpitak, A study of particle swarm technique for renewable energy power systems, in: Energy and Sustainable Development: Issues and Strategies (ESD), 2010 Proceedings of the International Conference on, IEEE, 2010, pp. 1–6.

[26] T. Logenthiran, D. Srinivasan, E. Phyu, Particle swarm optimization for demand side management in smart grid, in: Smart Grid Technologies - Asia (ISGT ASIA), 2015 IEEE Innovative, 2015, pp. 1–6.

[27] M. Manbachi, H. Farhangi, A. Palizban, S. Arzanpour, Smart grid adaptive energy conservation and optimization engine utilizing particle swarm optimization and fuzzification, Applied Energy 174 (2016) 69 – 79.

[28] M. Dorigo, Optimization, learning and natural algorithms, Ph. D. Thesis, Politecnico di Milano, Italy.

[29] A. Prakash, S. Deshmukh, A multi-criteria customer allocation problem in supply chain environment: an artificial immune system with fuzzy logic controller based approach, Expert Systems with Applications 38 (4) (2011) 3199–3208.

[30] HOMER (The Hybrid Optimization Model for Electric Renewables). Available from:, http://homerenergy.com/software.html.

[31] HYBRID2 (The Hybrid Power System Simulation Model). Available from:, http://www.ceere.org/rerl/projects/software/hybrid2/.

[32] iHOGA (improved Hybrid Optimization by Genetic Algorithms). Available from:, http://personal.unizar.es/rdufo/index.php.

[33] D. Connolly, H. Lund, B. Mathiesen, M. Leahy, A review of computer tools for analysing the integration of renewable energy into various energy systems, Applied Energy 87 (4) (2010) 1059 – 1082.

[34] J. V. Kadam, M. Schlegel, W. Marquardt, R. L. Tousain, D. H. van Hessem, J. H. van Den Berg, O. H. Bosgra, A two-level strategy of integrated dynamic optimization and control of industrial processes-a case study, Computer Aided Chemical Engineering 10 (2002) 511–516.

[35] J. Kadam, W. Marquardt, M. Schlegel, T. Backx, O. Bosgra, P. Brouwer, G. Dünnebier, D. Van Hessem, A. Tiagounov, S. De Wolf, Towards integrated dynamic real-time optimization and control of industrial processes, in: Foundations Of Computer-Aided Process Operations (FOCAPO2003), 2003, pp. 593–596.

[36] J. V. Kadam, W. Marquardt, Integration of economical optimization and control for intentionally transient process operation, in: Assessment and future directions of nonlinear model predictive control, Springer, 2007, pp. 419–434.

[37] L. T. Biegler, Technology advances for dynamic real-time optimization, Computer Aided Chemical Engineering 27 (2009) 1–6.

[38] A. Gopalakrishnan, L. T. Biegler, Economic nonlinear model predictive control for periodic optimal operation of gas pipeline networks, Computers & Chemical Engineering 52 (2013) 90–99.

[39] L. Würth, R. Hannemann, W. Marquardt, A two-layer architecture for economically optimal process control and operation, Journal of Process Control 21 (3) (2011) 311–321.

[40] X. Zhu, W. Hong, S. Wang, Implementation of advanced control for a heat-integrated distillation column system, in: 30th Annual Conference of IEEE Industrial Electronics Society (IECON), Vol. 3, 2004, pp. 2006–2011.

[41] J. Riccardi, M. Schiavetti, I. Fastelli, M. Cantù, Mathematical model of energy districts, Tech. Rep. INR RIC 2014 rev-02, ENEL Ingegneria e Ricerca (2014).

[42] P. Srikhirin, S. Aphornratana, S. Chungpaibulpatana, A review of absorption refrigeration technologies, Renewable and sustainable energy reviews 5 (4) (2001) 343–372.

[43] J. Nocedal, S. J. Wright, Numerical Optimization, 2nd Edition, Springer, 2006.